

Rajalakshmi Engineering College

Name: MADHUMITHA P
Email: 240701297@rajalakshmi.edu.in
Roll no: 240701297
Phone: 7904260818
Branch: REC
Department: I CSE AH
Batch: 2028
Degree: B.E - CSE

Scan to verify results



NeoColab_REC_CS23221_Python Programming

REC_Python_Week 4_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Imagine you are tasked with developing a function for calculating the total cost of an item after applying a sales tax. The sales tax rate is equal to 0.08 and it is defined as a global variable.

The function should accept the cost of the item as a parameter, calculate the tax amount, and return the total cost.

Additionally, the program should display the item cost, sales tax rate, and total cost to the user.

Function Signature: `total_cost(item_cost)`

Input Format

The input consists of a single line containing a positive floating-point number representing the cost of the item.

Output Format

The output consists of three lines:

"Item Cost:" followed by the cost of the item formatted to two decimal places.

"Sales Tax Rate:" followed by the sales tax rate in percentage.

"Total Cost:" followed by the calculated total cost after applying the sales tax, formatted to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00

Output: Item Cost: \$50.00

Sales Tax Rate: 8.0%

Total Cost: \$54.00

Answer

#

```
SALES_TAX_RATE = 0.08
def total_cost(item_cost):
    tax_amount = item_cost * SALES_TAX_RATE
    total_price = item_cost + tax_amount
    return total_price
item_cost = float(input())
total_price = total_cost(item_cost)

total_cost = total_cost(item_cost)
print(f"Item Cost: ${item_cost:.2f}")
print(f"Sales Tax Rate: {SALES_TAX_RATE * 100}%")
print(f"Total Cost: ${total_cost:.2f}")
```

Status : Correct

Marks : 10/10

2. Problem Statement

Amrita is developing a password strength checker for her website. She wants the checker to consider the length and the diversity of characters used in the password. A strong password should be long and include a mix of character types: uppercase, lowercase, digits, and special symbols.

She also wants the feedback to be user-friendly, so she wants to include the actual password in the output. Help Amrita finish this password checker using Python's built-in string methods.

Character Types Considered:

Lowercase letters (a-z) Uppercase letters (A-Z) Digits (0-9) Special characters (from string.punctuation, e.g. @, !, #, \$)

Input Format

The input consists of a single string representing the user's password.

Output Format

The program prints the strength of the password in this format:

If the password length < 6 characters or fewer than 2 of the 4 character types, the output prints "<password> is Weak"

If password length ≥ 6 and at least 2 different character types, the output prints "<password> is Moderate"

If Password length ≥ 10 and all 4 character types present, the output prints "<password> is Strong"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: password123

Output: password123 is Moderate

Answer

```
import string
```

```
def check_password_strength(password):  
    has_lower = any(char.islower() for char in password)  
    has_upper = any(char.isupper() for char in password)  
    has_digit = any(char.isdigit() for char in password)  
    has_special = any(char in string.punctuation for char in password)  
    types_present = sum([has_lower, has_upper, has_digit, has_special])  
    if len(password) < 6 or types_present < 2:  
        strength = "Weak"  
    elif len(password) >= 10 and types_present == 4:  
        strength = "Strong"  
    else:  
        strength = "Moderate"  
    print(f"{password} is {strength}")  
password=input()  
check_password_strength(password)
```

Status : Correct

Marks : 10/10

3. Problem Statement

Arjun is working on a mathematical tool to manipulate lists of numbers. He needs a program that reads a list of integers and generates two lists: one containing the squares of the input numbers, and another containing the cubes. Arjun wants to use lambda functions for both tasks.

Write a program that computes the square and cube of each number in the input list using lambda functions.

Input Format

The input consists of a single line of space-separated integers representing the list of input numbers.

Output Format

The first line contains a list of the squared values of the input numbers.

The second line contains a list of the cubed values of the input numbers.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1 2 3

Output: [1, 4, 9]

[1, 8, 27]

Answer

```
numbers = list(map(int, input().split()))
square = lambda x: x ** 2
cube = lambda x: x ** 3
squared_list = list(map(square, numbers))
cubed_list = list(map(cube, numbers))
print(squared_list)
print(cubed_list)
```

Status : Correct

Marks : 10/10

4. Problem Statement

Create a program for a mathematics competition where participants need to find the smallest positive divisor of a given integer n . Your program should efficiently determine this divisor using the `min()` function and display the result.

Input Format

The input consists of a single positive integer n , representing the number for which the smallest positive divisor needs to be found.

Output Format

The output prints the smallest positive divisor of the input integer in the format: "The smallest positive divisor of $[n]$ is: [smallest divisor]".

Refer to the sample output for the exact format.

Sample Test Case

Input: 24

Output: The smallest positive divisor of 24 is: 2

Answer

```
def smallest_divisor(n):  
    divisor = min([i for i in range(2, n+1) if n % i == 0])  
    print(f"The smallest positive divisor of {n} is: {divisor}")  
n=int(input())  
smallest_divisor(n)
```

Status : Correct

Marks : 10/10